IN THE CLAIMS

1. (Previously Presented) A method comprising:

providing prerequisite information regarding page sub-components of a graphical user interface (GUI) that are prerequisites to other page sub-components of the GUI;

in response to a request for a destination page of the GUI and with reference to the prerequisite information, instantiating a container object corresponding to the requested destination page of the GUI;

the container object identifying one or more page sub-component prerequisites of page sub-components associated with the destination page by invoking an execution of a method of the container object;

the container object determining whether one or more requirements of an instance of an identified page sub-component prerequisite have been satisfied by invoking a method of a component object corresponding to the instance that causes stored information regarding the state of the page sub-component prerequisite to be retrieved from a current component object, the component object being a child object of the container object; and

causing the destination page to be displayed with (1) content associated with the identified page sub-component prerequisite if any of its one or more requirements have not been satisfied and (2) content associated with those of the page sub-components having no page sub-component prerequisites or having page sub-component prerequisites whose requirements have all been satisfied by invoking an execution of a method of the corresponding component object to stream the content to the container object.

2. (Previously Presented) The method of claim 1 wherein the prerequisite information is stored in a Java properties file, wherein the method further comprises:

the container object instantiating a page sub-component prerequisite object using information stored in the Java properties file; and

retrieving the prerequisite information by executing a method of the page sub-component prerequisite object, wherein the page sub-component prerequisite object streams the prerequisite information to the container object in response to the execution.

3. (Previously Presented) The method of claim 2, further comprising supporting definition of hierarchical relationships of page sub-component prerequisites by:

iterating through each of the identified page sub-component prerequisites for a particular page sub-component associated with the destination page in a predetermined order by executing a method of the corresponding page sub-component prerequisite object until encountering the first page sub-component prerequisite that has one or more unsatisfied requirements; and

displaying the first page sub-component prerequisite of the identified page sub-component prerequisites before displaying a second page sub-component prerequisite of the identified page sub-component prerequisites that has one or more unsatisfied requirements and that is dependent upon the first page sub-component prerequisite according to the predetermined order.

4. (Previously Presented) The method of claim 2, wherein each page sub-component has a prerequisite property stored in the Java properties file and the prerequisite information

includes, for each page sub-component that has one or more page sub-component prerequisites, a string identifying the one or more page sub-component prerequisites.

5. (Previously Presented) The method of claim 4, wherein the prerequisite information is structured as a list of attribute-value pairs in the Java properties file, and wherein a syntax for identifying a first sub-component, $sub_1$, and a second sub-component, $sub_2$, as prerequisites of a third sub-component, $sub_3$, is substantially as follows:

$sub_3$.prereq = $sub_1$ $sub_2$.

6. (Original) The method of claim 1, wherein the request for the destination page comprises a HyperText Transfer Protocol (HTTP) request, and wherein the page sub-components of the GUI are associated with web pages.

7. (Previously Presented) The method of claim 2, further comprising modifying the prerequisite information via the Java properties file without necessitating recompilation of software code corresponding to the component object.

8. (Previously Presented) The method of claim 1, wherein said determining whether one or more requirements of an instance of an identified page sub-component prerequisite have been satisfied includes requesting that a page sub-component prerequisite object verify whether all its requirements have been satisfied by invoking an execution of a method of the page sub-component prerequisite object.

9. (Previously Presented) The method of claim 1, wherein page sub-component objects corresponding to the page sub-components of the GUI and page sub-component prerequisite objects responsible for ensuring satisfaction of one or more prerequisite conditions are loosely coupled and are dynamically associated with each other by way of the prerequisite information.

10. (Previously Presented) A method of presenting a page requested by a user comprising:

in response to a request for a destination page of a graphical user interface (GUI), creating an instance of a container to represent the destination page, the container including a list of sub-components to render;

identifying one or more sub-components associated with the destination page by executing a method of the container;

for each of the one or more sub-components, determining whether the sub-component has any page sub-component prerequisites with reference to a set of prerequisite information, the set of prerequisite information including information regarding sub-components of the GUI that are prerequisites to other sub-components of the GUI, and

if the sub-component has a page sub-component prerequisite and if one or more requirements of the page sub-component prerequisite remains unsatisfied, then adding an instance of the page sub-component prerequisite to the list of sub-components associated with the container, otherwise adding an instance of the sub-component to the list of sub-components; and

causing the destination page to be displayed by rendering the output of the instances on the list of sub-components, whereby page sub-component prerequisites that have one or more requirements that remain unsatisfied are displayed in place of the corresponding sub-

components, wherein the rendering is performed by invoking an execution of a method of an object corresponding to an instance of the list of sub-components associated with the container.

11. (Previously Presented) A graphical user interface (GUI) system for enforcing page sub-component prerequisites comprising:

a properties data store including information regarding page sub-components of the GUI that are prerequisites to other page sub-components of the GUI;

a base agent to respond to requests for a destination page of the GUI, in response to a request for the destination page, the base agent creating an instance of a container to represent the destination page and initiating display of the destination page after a list of page sub-components of the container has been populated by invoking an execution of a method of an object corresponding to each page sub-component of the list in the container; and

a sub-component prerequisite factory decoupling the page sub-components from their respective page sub-component prerequisites, the sub-component prerequisite factory to either (1) cause an instance of an identified page sub-component prerequisite to be added to the list of page sub-components if it determines that one or more requirements of the identified page sub-component prerequisite are unsatisfied or (2) cause an instance of the page sub-component to be added to the list of page sub-components, whereby page sub-component prerequisites that have one or more unsatisfied requirements are displayed in place of the corresponding page sub-components.

12. (Previously Presented) The system of claim 11, wherein the properties data store is a Java properties file, wherein the container is to

instantiate a page sub-component prerequisite object using information stored in the Java properties file, and

execute a method of the page sub-component prerequisite object to cause the page sub-component prerequisite object to stream the prerequisite information to the container.

13. (Previously Presented) The system of claim 12, wherein the sub-component prerequisite factory supports hierarchical relationships of page sub-component prerequisites by:

iterating through each of the identified page sub-component prerequisites for a particular page sub-component associated with the destination page in a predetermined order by executing a method of the corresponding page sub-component prerequisite object until encountering the first page sub-component prerequisite that has one or more unsatisfied requirements; and

displaying the first page sub-component prerequisite of the identified page sub-component prerequisites before displaying a second page sub-component prerequisite of the identified page sub-component prerequisites that has one or more unsatisfied requirements and that is dependent upon the first page sub-component prerequisite according to the predetermined order.

14. (Previously Presented) The system of claim 12, wherein each page sub-component has a prerequisite property stored in the Java properties file and the prerequisite information includes, for each page sub-component that has one or more prerequisite sub-components, a string identifying the one or more page sub-component prerequisites.

15. (Previously Presented) The system of claim 14, wherein at least a portion of the information of the properties data store is structured as a list of attribute-value pairs in the Java properties file, and wherein a syntax for identifying a first sub-component, $sub_1$, and a second sub-component, $sub_2$, as prerequisites of a third sub-component, $sub_3$, is substantially as follows:

$sub_3.prereq = sub_1\ sub_2$.

16. (Original) The system of claim 11, wherein the requests correspond to HyperText Transfer Protocol (HTTP) requests, and wherein the page sub-components of the GUI are associated with web pages.

17. (Previously Presented) The system of claim 11, wherein prerequisite relationships among two or more page sub-components of the page sub-components of the GUI are modified without necessitating recompilation of software code corresponding to the page sub-components by editing the information of the properties data store independently.

18. (Previously Presented) The system of claim 11, further comprising page sub-component objects corresponding to the page sub-components of the GUI and page sub-component prerequisite objects responsible for ensuring satisfaction of one or more prerequisite conditions are loosely coupled and are dynamically associated with each other by way of the prerequisite information by invoking an execution of a method of the corresponding page sub-component prerequisite object.

19. (Previously Presented) A machine-readable medium having stored thereon data representing sequences of instructions, the sequences of instruction which, when executed by a processor, cause the processor to:

identify one or more page sub-component prerequisites of page sub-components associated with a destination page of a graphical user interface (GUI) in response to a request for the destination page and with reference to the prerequisite information regarding page sub-components of the GUI that are prerequisites to other page sub-components of the GUI;

determine whether one or more requirements of an instance of an identified page sub-component prerequisite have been satisfied by invoking a method of the instance that causes stored information regarding the state of the page sub-component prerequisite to be retrieved from a current context; and

cause the destination page to be displayed with (1) content associated with the identified page sub-component prerequisite if any of its one or more requirements have not been satisfied and (2) content associated with those of the page sub-components having no page sub-component prerequisites or having page sub-component prerequisites whose requirements have all been satisfied.

20. (Previously Presented) The method of claim 1, wherein at least one component object comprises one or more sub-container objects, and wherein at least one of the sub-container objects includes one or more sub-components.

21. (Previously Presented) The method of claim 20, wherein the container objects and the component objects are maintained in a hierarchical structure, wherein the component objects are implemented within one or more leaf nodes, and wherein the container objects are

implemented within one or more non-leaf nodes, each non-leaf node having at least one of a leaf sub-node and a non-leaf sub-node.

22. (Previously Presented) The method of claim 21, wherein a container object represents a page and a component object represent content of the page including at least one of the following:

a chart;

a table;

a scroll list; and

a data entry.

23. (Previously Presented) The method of claim 4, wherein a first sub-component is identified as a prerequisite of a second sub-component by assigning the first sub-component to the second sub-component having the string identifying the prerequisites appended to the second sub-component separated by a delimiter.

24. (Previously Presented) The method of claim 23, wherein the prerequisites of the second sub-component is inherited from the first sub-component.

25. (Previously Presented) The method of claim 24, wherein at least one prerequisite of the second sub-component used in a context is overridden by assigning an overriding value to the second sub-component having an identity of the context prefixed to an identity of the second sub-component separated by a delimiter.